**Section: B**

**Name : Muhammad usama.**

**Roll no:17f-8195**

**Lab :10**

**Task:1**

#include<iostream>

#include<string>

using namespace std;

class employee

{

string name;

int salary;

public:

employee()

{

cout<<"enetr salary of employee : ";

cin>>salary;

}

void show()

{

cout<<"salary : "<<salary<<endl;

}

friend employee &operator +=(employee &obj, int var);

};

employee &operator +=(employee &obj, int var)

{

obj.salary=obj.salary+var;

return obj;

}

int main()

{

employee obj;

int add;

cout<<"enetr amount that you want to add : ";

cin>>add;

obj+=add;

cout<<"your salary now is : ";

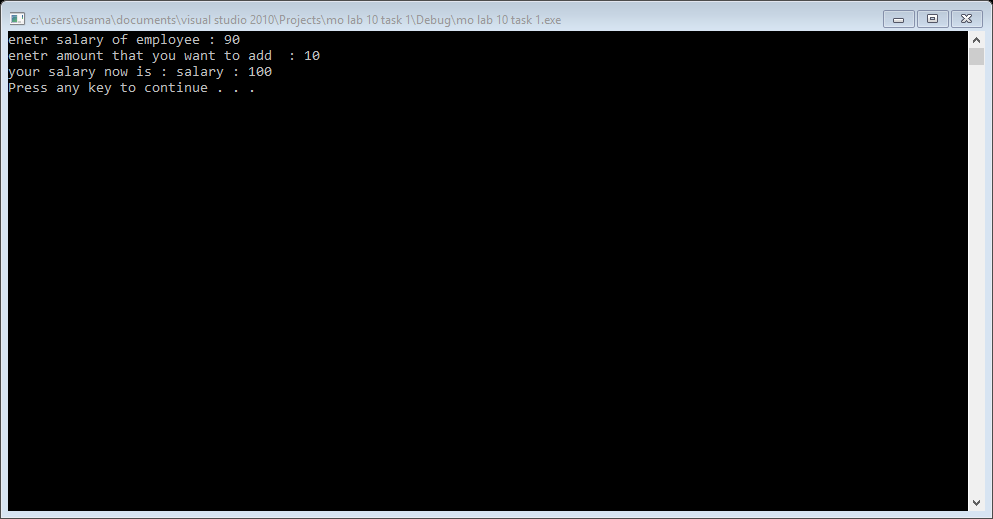
obj.show();

system("pause");

return 0;

}

**Result:**

****

**Task 2:**

#include<iostream>

#include<cmath>

#include<ymath.h>

#include<math.h>

using namespace std;

class point

{

int x;

int y;

public:

point()

{

x=0;

y=0;

}

void show()

{

cout<<"x : "<<x<<endl;

cout<<"y : "<<x<<endl;

}

void get()

{

cout<<"enter x : ";

cin>>x;

cout<<"enter y : ";

cin>>y;

}

friend point &operator +=(point &obj1,int var);

friend point &operator ==(point &obj1,point &obj2);

friend point &operator !(point &obj1);

friend float &operator /(point &obj1,point obj2);

};

point &operator +=(point &obj1,int var)

{

obj1.x=obj1.x+var;

obj1.y=obj1.y+var;

return obj1;

}

point &operator ==(point &obj1,point &obj2)

{

if(obj1.x==obj2.x)

{

if(obj1.y==obj2.y)

{

cout<<"this two points are equals"<<endl;

}

else

{

cout<<"these points are not equals"<<endl;

}

}

else

{

cout<<"this two pints are not equal"<<endl;

}

return obj2;

}

point &operator !(point &obj)

{

point temp;

temp.x=obj.x;

obj.x=obj.y;

obj.y=temp.x;

return obj;

}

float &operator /(point &obj1,point obj2)

{

point temp;

float x;

float result;

temp.x=obj1.x-obj2.x;

temp.y=obj1.y-obj2.y;

temp.x=temp.x\*temp.x;

temp.y=temp.y\*temp.y;

x=temp.x+temp.y;

// cout<<x<<endl;

result=sqrt(x);

return result;

}

void add();

void equallity();

void swaping();

void distance();

int main()

{

int choice=0;

do

{

cout<<"1 : for addition"<<endl;

cout<<"2 : for equallity"<<endl;

cout<<"3 : for distance "<<endl;

cout<<"4 : for swaping"<<endl;

cout<<"5 : for exit"<<endl;

cin>>choice;

system("cls");

if(choice==1)

{

add();

}

if(choice==2)

{

equallity();

}

if(choice==3)

{

distance();

}

if(choice==4)

{

swaping();

}

}while(choice!=5);

system("pause");

return 0;

}

void add()

{

point obj1,obj2;

int v;

int u;

cout<<"enetr values of obj1"<<endl;

obj1.get();

cout<<"enter value that you want to add in obj1 : ";

cin>>v;

obj1+=v;

cout<<"enetr values for obj2"<<endl;

obj2.get();

cout<<"enter value that you want to add in obj2 : ";

cin>>u;

obj2+=u;

cout<<"value of obj1 after add a number"<<endl;

obj1.show();

cout<<"value of obj2 after add a number"<<endl;

obj2.show();

}

void equallity()

{

point obj1,obj2;

cout<<"enetr values of obj1"<<endl;

obj1.get();

cout<<"enetr values for obj2"<<endl;

obj2.get();

obj1==obj2;

}

void swaping()

{

point obj1,obj2;

cout<<"enetr values of obj1"<<endl;

obj1.get();

cout<<"enetr values for obj2"<<endl;

obj2.get();

!obj1,obj2;

cout<<"value of obj1 "<<endl;

obj1.show();

cout<<"value of obj2"<<endl;

obj2.show();

}

void distance()

{

point obj1,obj2;

float result;

cout<<"enetr values of obj1"<<endl;

obj1.get();

cout<<"enetr values for obj2"<<endl;

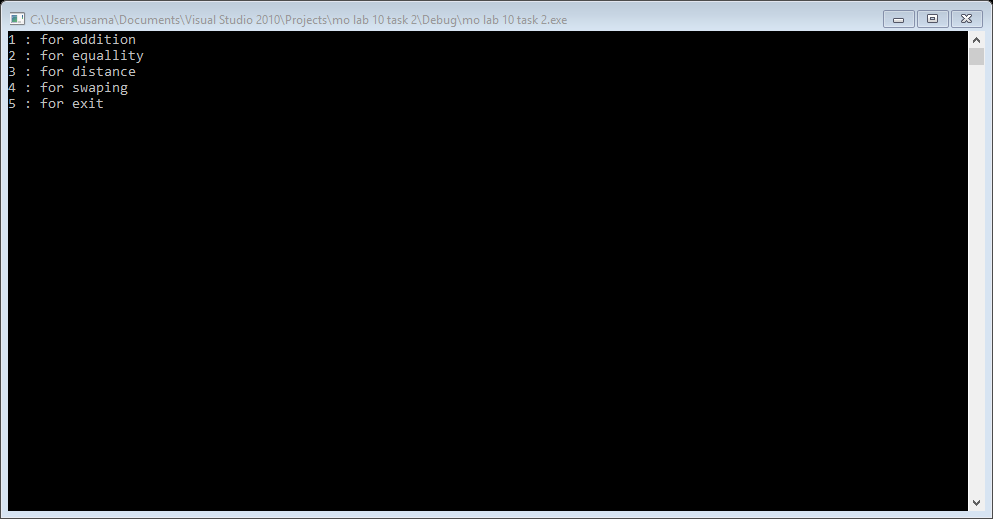
obj2.get();

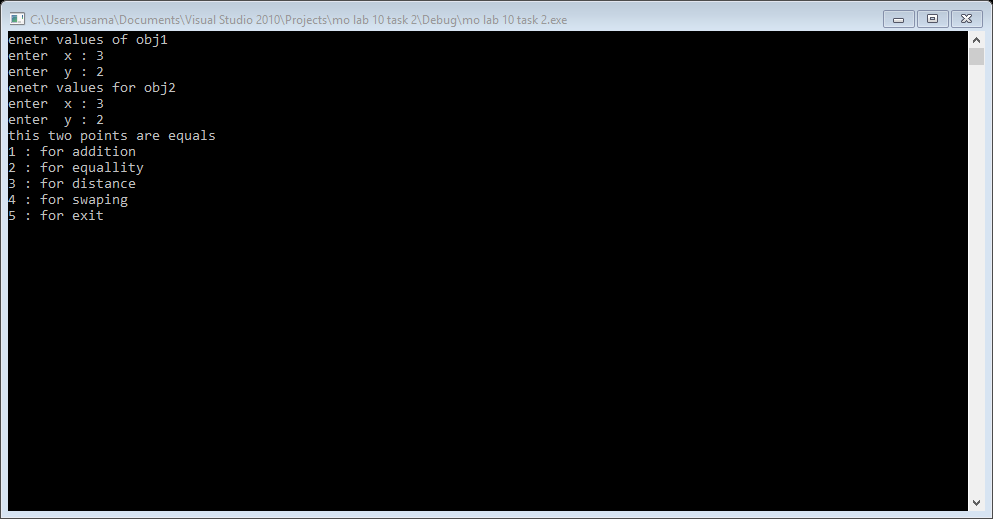
result=obj1/obj2;

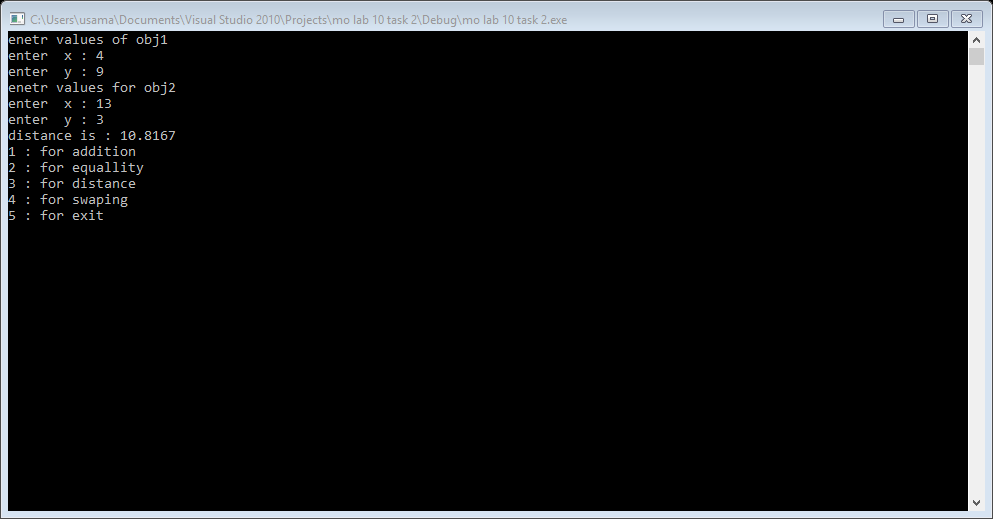
cout<<"distance is : "<<result<<endl;

}

**Result:**

****

****

****

**Task 3:**

#include<iostream>

using namespace std;

class matrix

{

int \*\*ptr1;

int r;

int c;

public:

matrix()

{

//ptr1[2][2];

ptr1=new int\*[2];

for(int i=0; i<2; i++)

{

ptr1[i]=new int[2];

}

}

void getrows()

{

cout<<"enetr row : ";

cin>>r;

cout<<"enetr column : ";

cin>>c;

}

void get()

{

cout<<"enetr values for matrix "<<endl;

for(int i=0; i<2; i++)

{

for(int j=0; j<2; j++)

{

cout<<"row : "<<i+1 <<" column : "<<j+1<<endl;

cin>>ptr1[i][j];

}

}

}

void show()

{

for(int i=0; i<2; i++)

{

for(int j=0; j<2; j++)

{

cout<<ptr1[i][j]<<" ";

}

cout<<endl;

}

}

friend void operator +(matrix &obj1,matrix &obj2);

friend void operator ==(matrix &obj1,matrix &obj2);

~matrix()

{

for(int i=0; i<2; i++)

{

delete []ptr1[i];

}

delete []ptr1;

ptr1=NULL;

}

};

void operator +(matrix &obj1,matrix &obj2)

{

int \*\*temp;

temp=new int\*[2];

for(int i=0; i<2; i++)

{

temp[i]=new int[2];

}

/\*for(int i=0; i<2; i++)

{

for(int j=0; j<2; j++)

{

temp[i][j]=0;

}

}\*/

for(int i=0; i<2; i++)

{

for(int j=0; j<2; j++)

{

temp[i][j]=obj1.ptr1[i][j]+obj2.ptr1[i][j];

}

}

cout<<"resullt of addition of two matrix "<<endl;

for(int i=0; i<2; i++)

{

for(int j=0; j<2; j++)

{

cout<<temp[i][j]<<" ";

}

cout<<endl;

}

}

void operator ==(matrix &obj1,matrix &obj2)

{

if(obj1.c==obj2.c && obj1.r==obj2.r)

{

cout<<"the matrix have same order "<<endl;

}

else

{

cout<<"matrix does not have same order "<<endl;

}

}

void addition();

void matrix\_order();

int main()

{

int choice;

do

{

cout<<"1 : for addition "<<endl;

cout<<"2 : for check matrix order"<<endl;

cout<<"3 : for terminate programe"<<endl;

cin>>choice;

system("cls");

if(choice==1)

{

addition();

}

if(choice==2)

{

matrix\_order();

}

}while(choice!=3);

system("pause");

return 0;

}

void addition()

{

matrix obj1,obj2;

cout<<"enter values for matrix 1"<<endl;

obj1.get();

cout<<"enter values for matrix 2"<<endl;

obj2.get();

cout<<"you enter first matrix is "<<endl;

obj1.show();

cout<<"you enter second matrix is "<<endl;

obj2.show();

obj1+obj2;

}

void matrix\_order()

{

matrix obj1,obj2;

cout<<"enter rows and columns for first matrix"<<endl;

obj1.getrows();

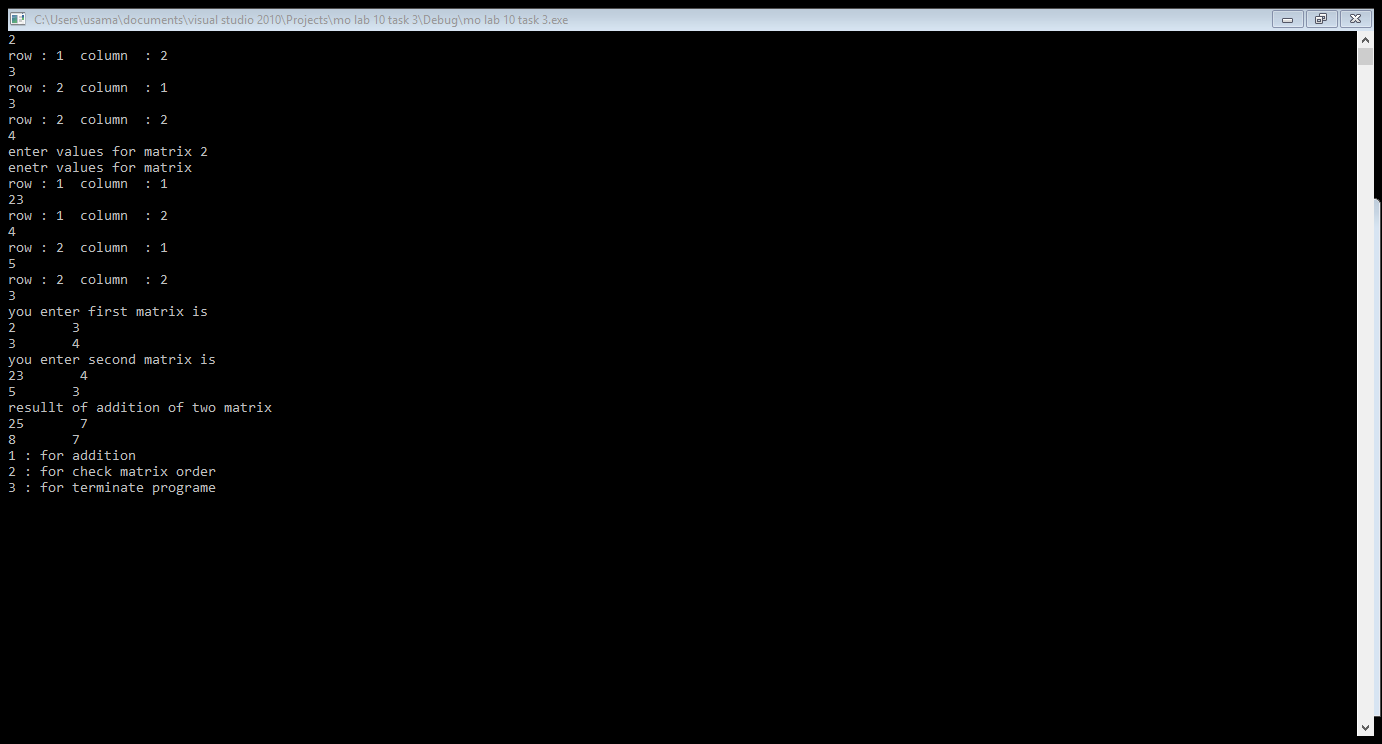
cout<<"enter rows and columns for second matrix"<<endl;

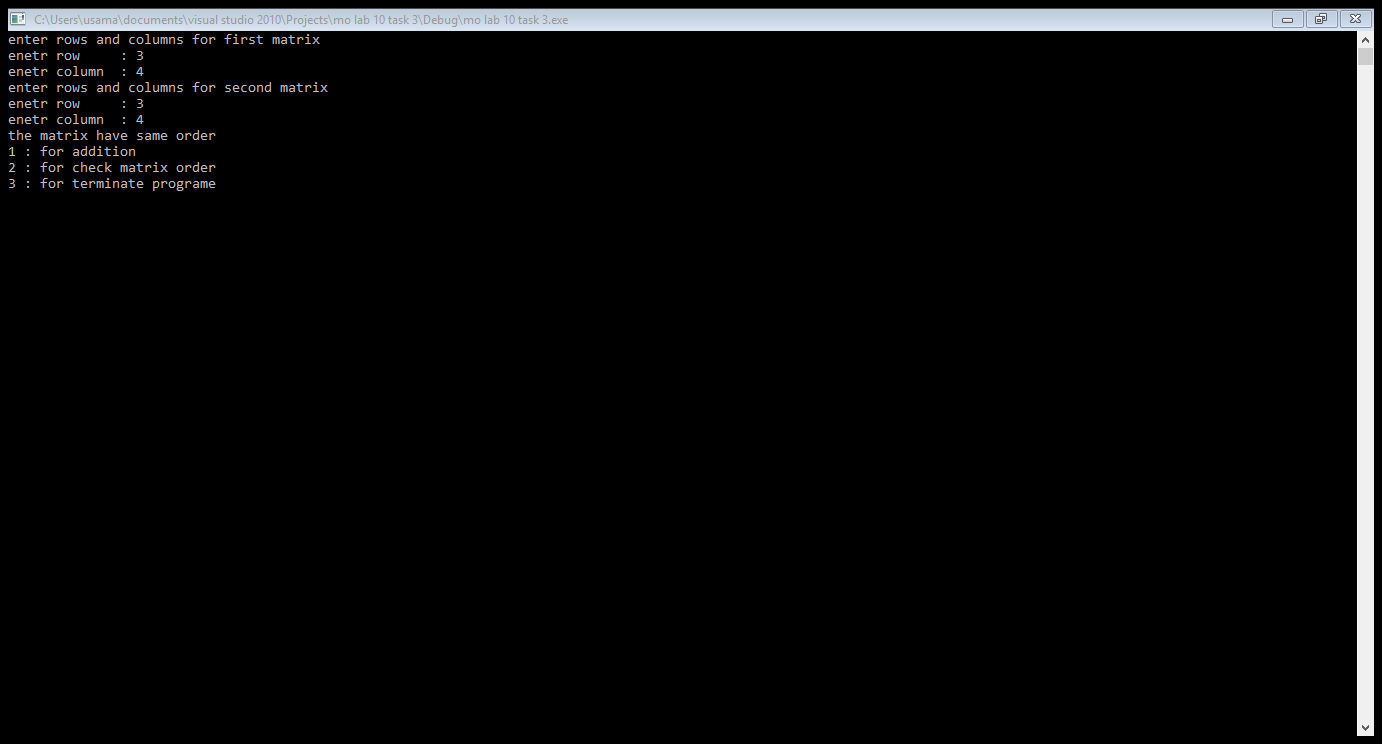
obj2.getrows();

obj1==obj2;

}

**Result:**

****

****

**Task 4:**

#include<iostream>

using namespace std;

class rational

{

float x;

float y;

public:

void show()

{

cout<<"x : "<<x<<endl;

cout<<"y : "<<y<<endl;

}

void get()

{

cout<<"enter value of x :";

cin>>x;

cout<<"enter value of y :";

cin>>y;

}

rational &operator +(rational &obj1)

{

rational temp;

temp.x=x+obj1.x;

temp.y=y+obj1.y;

return temp;

}

rational operator \*(rational &obj1)

{

rational temp;

temp.x=x\*obj1.x;

temp.y=y\*obj1.y;

return temp;

}

rational operator -(rational &obj1)

{

rational temp;

temp.x=x-obj1.x;

temp.y=y-obj1.y;

return temp;

}

rational operator /(rational &obj1)

{

rational temp;

temp.x=x/obj1.x;

temp.y=y/obj1.y;

return temp;

}

};

void addition();

void multiplication();

void subtraction();

void devison();

int main()

{

int choice;

do

{

cout<<"1 : for addition"<<endl;

cout<<"2 : for subtraction"<<endl;

cout<<"3 : for multiplication"<<endl;

cout<<"4 : for devision"<<endl;

cout<<"5 : for exit"<<endl;

cin>>choice;

system("cls");

if(choice==1)

{

addition();

}

if(choice==2)

{

subtraction();

}

if(choice==3)

{

multiplication();

}

if(choice==4)

{

devison();

}

}while(choice!=5);

system("pause");

return 0;

}

void addition()

{

rational obj1,obj2,obj3;

cout<<"enter value for obj1 "<<endl;

obj1.get();

cout<<"enter value for obj2 "<<endl;

obj2.get();

obj3=obj2+obj1;

cout<<"reesult of addtion of two objects "<<endl;

obj3.show();

cout<<endl;

}

void multiplication()

{

rational obj1,obj2,obj3;

cout<<"enter value for obj1 "<<endl;

obj1.get();

cout<<"enter value for obj2 "<<endl;

obj2.get();

obj3=obj2\*obj1;

cout<<"result of multiplication of two objects "<<endl;

obj3.show();

cout<<endl;

}

void subtraction()

{

rational obj1,obj2,obj3;

cout<<"enter value for obj1 "<<endl;

obj1.get();

cout<<"enter value for obj2 "<<endl;

obj2.get();

obj3=obj2-obj1;

cout<<"result of subtraction of two objects "<<endl;

obj3.show();

cout<<endl;

}

void devison()

{

rational obj1,obj2,obj3;

cout<<"enter value for obj1 "<<endl;

obj1.get();

cout<<"enter value for obj2 "<<endl;

obj2.get();

obj3=obj2/obj1;

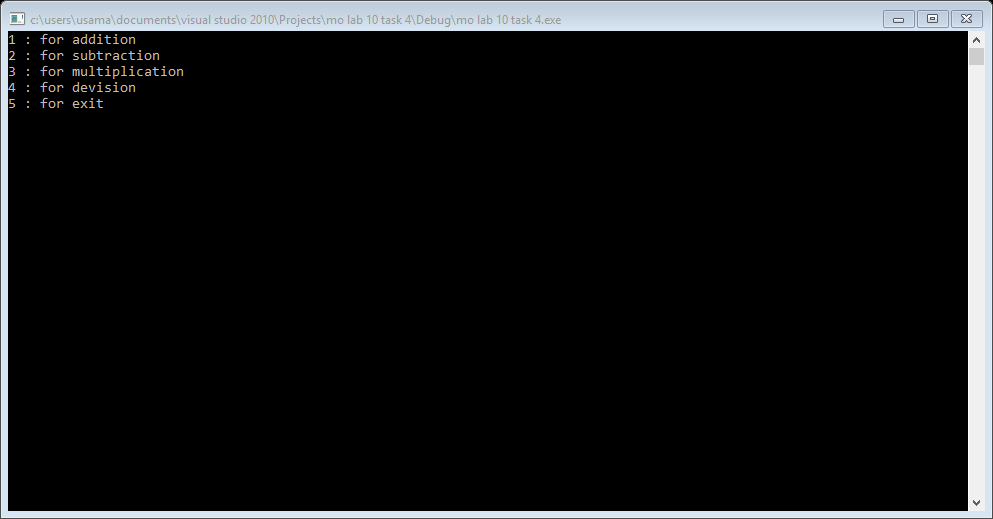
cout<<"result of devision of two objects "<<endl;

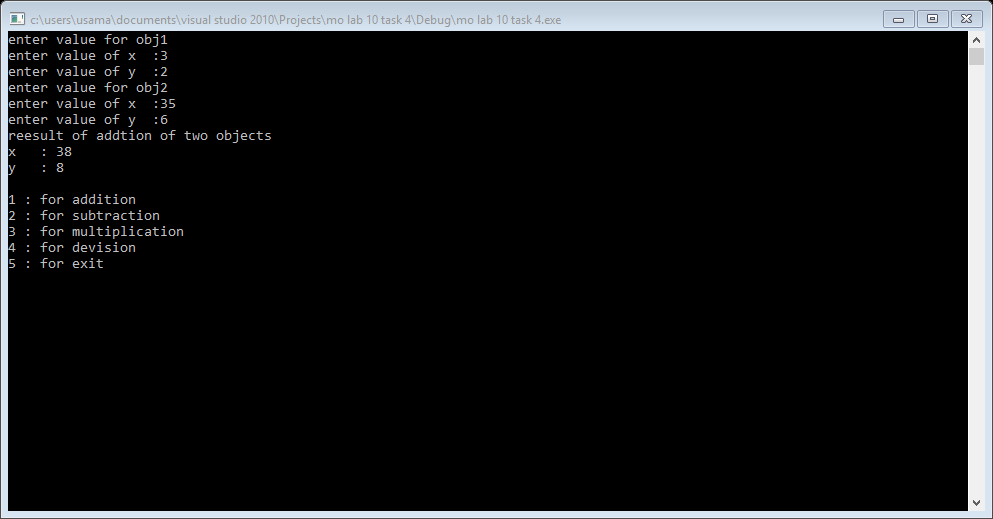
obj3.show();

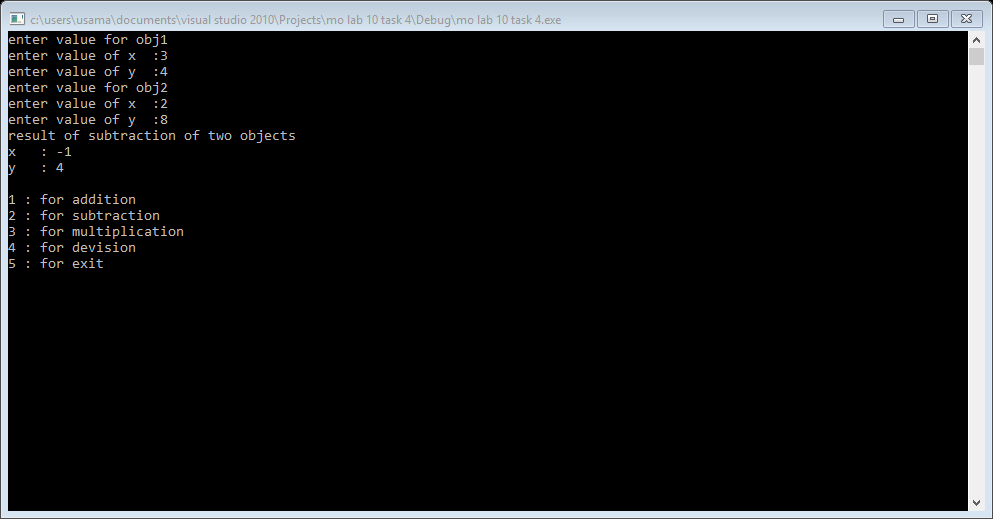
cout<<endl;

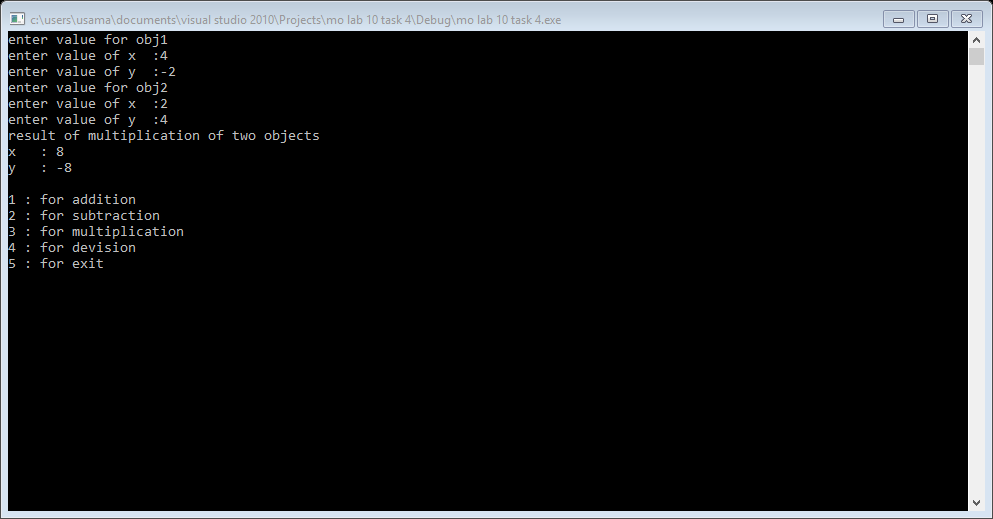
}

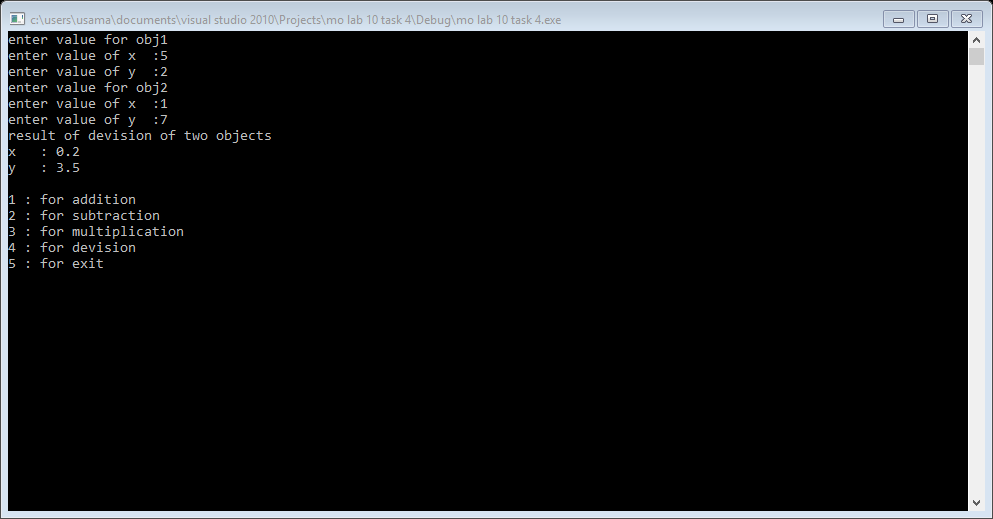
**Result:**

****

****

****

****

****

**Task 5:**

#include<iostream>

using namespace std;

class factorial

{

int f;

public:

void get()

{

cout<<"enetr any no : ";

cin>>f;

}

friend void operator !(factorial &obj);

};

void operator !(factorial &obj)

{

int x=1;

int y=1;

for(int x=1; x<=obj.f; x++)

{

y=y\*x;

}

cout<<"foctorial of no : "<<y<<endl;

}

int main()

{

factorial obj;

obj.get();

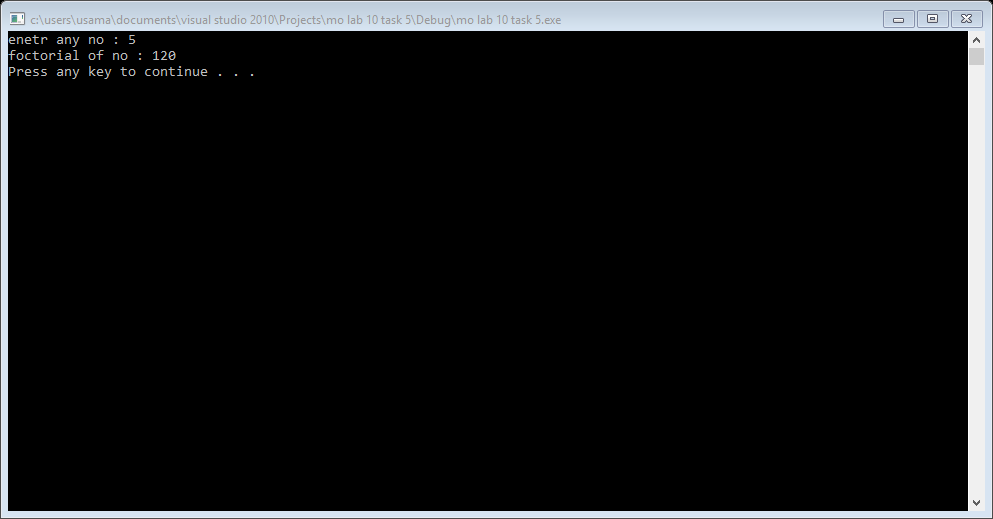
!obj;

system("pause");

return 0;

}

**Result:**

****